

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2003-15522  
(P2003-15522A)

(43) 公開日 平成15年1月17日 (2003.1.17)

(51) Int.Cl.<sup>7</sup>

G 0 9 C 1/00

識別記号

6 1 0

F I

G 0 9 C 1/00

テ-マ-ト\*(参考)

6 1 0 A 5 J 1 0 4

審査請求 未請求 請求項の数 5 O L (全 22 頁)

(21) 出願番号 特願2001-195752(P2001-195752)

(22) 出願日 平成13年6月28日(2001.6.28)

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番  
1号

(72) 発明者 岡田 壮一

神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(72) 発明者 鳥居 直哉

神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(74) 代理人 100094145

弁理士 小野 由己男 (外2名)

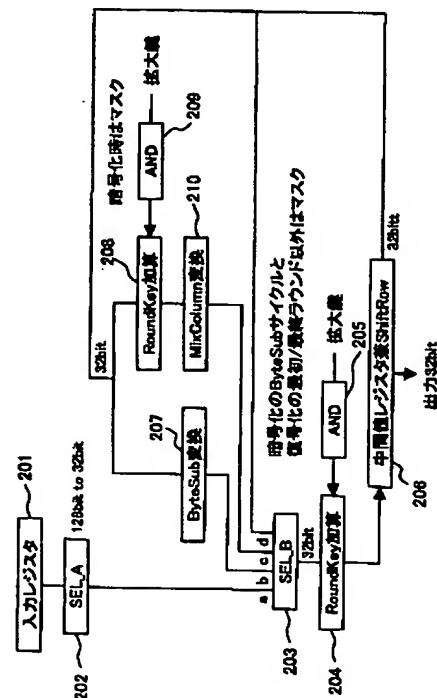
最終頁に続く

(54) 【発明の名称】 暗号回路

(57) 【要約】

【課題】 AESブロック暗号を実装する際に、回路規模を小型化するとともに一定水準の高速処理が実現可能な暗号回路を提供する。

【解決手段】 Round処理部が、Round Keyの値を入力データに加算する第1 Round Key加算回路と、第1 Round Key加算回路の出力を一時的に格納するとともにShift Row変換を実行する中間レジスタ兼Shift Row変換回路と、中間レジスタ兼Shift Row変換回路の値が入力されByte Sub変換を実行するByte Sub変換回路と、中間レジスタ兼Shift Row変換回路の値が入力されRound Keyの値を加算する第2 Round Key加算回路と、第2 Round Key加算回路の出力に対してMix Column変換を実行するMix Column変換回路と、第1セクタ、中間レジスタ兼Shift Row変換回路、Byte Sub変換回路、Mix Column変換回路の出力のうちいずれか1つを第2 Round Key加算回路に出力する第2セクタとを備える。



BEST AVAILABLE COPY

## 【特許請求の範囲】

【請求項 1】暗号鍵から所定の処理ブロック長に対応するビット数の複数の Round Key を生成し、入力データと前記 Round Key のうちの 1 つと排他的論理和演算を実行する EXOR 演算部と、Byte Sub 変換、Shift Row 変換、Mix Column 変換、Round Key 加算を含む Round 処理を複数回実行する Round 処理部とを備えるラウンドファンクション部により入力データと前記 Round Key による暗号化／復号化処理を前記処理ブロック長毎に実行する暗号回路であって、

前記 Round 処理部が、入力データを前記処理ブロック長よりも小さい実行ブロック長に分割する第 1 セクタと、前記実行ブロック長毎に前記 Round Key の値を入力データに加算する第 1 Round Key 加算回路と、前記第 1 Round Key 加算回路の出力を一時的に格納するとともに前記処理ブロック長による Shift Row 変換を実行する中間レジスタ兼 Shift Row 変換回路と、前記中間レジスタ兼 Shift Row 変換回路の値が前記実行ブロック長毎に入力され Byte Sub 変換を実行する Byte Sub 変換回路と、前記中間レジスタ兼 Shift Row 変換回路の値が前記実行ブロック長毎に入力され前記 Round Key の値を前記実行ブロック長毎に加算する第 2 Round Key 加算回路と、前記第 2 Round Key 加算回路の出力に対して Mix Column 変換を実行する Mix Column 変換回路と、前記第 1 セクタ、中間レジスタ兼 Shift Row 変換回路、Byte Sub 変換回路、Mix Column 変換回路の出力のうちいずれか 1 つを前記第 2 Round Key 加算回路に出力する第 2 セクタとを備えることを特徴とする暗号回路。

【請求項 2】前記 Byte Sub 変換回路は、入力データに対して行列演算を実行する復号化用行列演算部と、入力データと前記復号化用行列演算部の出力のうちいずれかを出力する第 3 セクタと、前記第 3 セクタから出力されるデータに対して逆数演算を実行する逆数演算部と、前記逆数演算部から出力されるデータに対して行列演算を実行する暗号化用行列演算部と、前記逆数演算部の出力と前記暗号化用行列演算部の出力のうちいずれか一方を出力する第 4 セクタとを備える請求項 1 に記載の暗号回路。

【請求項 3】前記中間レジスタ兼 Shift Row 変換回路に入力されるデータのシフト量に関するシフトデータを暗号化時と復号化時において逆順に入力することにより、前記中間レジスタ兼 Shift Row 変換回路を暗号化と復号化に共通に使用する、請求項 1 または 2 に記載の暗号回路。

【請求項 4】前記 Mix Column 変換回路は、乗数固定の複数の乗算器と前記複数の乗算器の排他的論理和を演算する EXOR 回路とを備え、各乗算器に入力されるデータと各乗算器に設定された乗数との間で行列演算を実行する、請求項 1 ～ 3 のいずれかに記載の暗号回路。

【請求項 5】暗号鍵を前記実行ブロック長に応じたビッ

ト数に分割して出力する第 5 セクタと、

前記実行ブロック長毎にデータをラッチするフリップフロップ回路が複数段接続されたシフトレジスタと、

前記シフトレジスタの最終段のフリップフロップ回路の出力と定数群のうちから選択される 1 つの定数との排他的論理和演算を実行する第 1 EXOR 回路と、

前記シフトレジスタのフリップフロップのうち暗号化時に演算対象となるフリップフロップの出力と復号化時に演算対象となるフリップフロップの出力とが入力されい

10 づれか 1 つを選択的に出力する第 6 セクタと、

前記第 6 セクタの出力をローテーション処理する Rot Byte 処理回路と、

前記第 6 セクタの出力と前記 Rot Byte 処理回路の出力とが入力され、いずれか 1 つを選択的に出力する第 7 セクタと、

前記第 7 セクタの出力に対して前記実行ブロック長毎の Byte Sub 変換を実行する Sub Byte 処理回路と、

前記第 6 セクタの出力と前記 Sub Byte 処理回路の出力が入力され、いずれか 1 つを選択的に出力する第 8 セクタと、

20 前記第 1 EXOR 回路の出力と前記第 8 セクタの出力とに基づいて排他的論理和演算を実行する第 2 EXOR 回路と、前記シフトレジスタのフリップフロップのうち暗号化時にその出力が演算対象となるフリップフロップに対して、前記第 2 EXOR 回路の出力と、隣接する段のフリップフロップの出力とのうちいずれかを選択的に出力するシフトレジスタ部セクタと、を備える、前記暗号鍵から所定の処理ブロック長に対応するビット数の複数の Round Key を前記実行ブロック長に対応するビット数毎に分割された拡大鍵として生成する拡大鍵スケジュール回路を有する請求項 1 ～ 4 のいずれかに記載の暗号回路。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、共通鍵暗号として現在の米国標準である DES に代わり、次世代の共通鍵ブロック暗号 AES (Advanced Encryption Standard) として標準化されつつある Rijndael アルゴリズムをハードウェア実装するための暗号回路に関する。

## 【0002】

40 【従来の技術】電子商取引や電子マネー等、インターネットを用いた様々なサービスが検討されている。これらの技術は、個人の生活にとどまらず、会社間の取引や生産性の向上等様々な分野で広く用いられていく技術である。特に、今後は、個人認証を目的として、スマートカードやモバイル機に暗号機能が搭載され、認証、電子署名、及びデータの暗号化等に広く使われると予想される。

【0003】これらの応用において、インターネット上の第三者からの盗聴を防ぐ意味で共通鍵暗号が用いられる。共通鍵暗号は、現在の米国標準である DES に代わる

次世代の共通鍵ブロック暗号AES (Advanced Encryption Standard) として、Rijndaelと呼ばれる暗号アルゴリズムが選定され、標準化されようとしている (AESドラフト <http://csrc.nist.gov/publications/drafts/drafts-AES.pdf>参照)。

【0004】AESは処理ブロック長128ビットのブロック暗号であり、その暗号化アルゴリズムは、図1に示すように、ラウンドファンクション部20と鍵スケジュール部10で構成される暗号回路により実現可能であると考えられる。ラウンドファンクション部20は、入力データを一時格納する入力レジスタ21と、入力データと拡大鍵との排他的論理和演算を行うEXOR処理部22とRound処理部23、最終Round処理部24および出力データを一時格納する出力レジスタ25からなる。

【0005】Round処理部23は、Byte Sub変換部31、Shift Row変換部32、Mix Column変換部33およびRound Key加算部34からなり、最終Round処理部24はRound処理部23からMix Column変換部33を除いた処理を行うもので、Byte Sub変換部35、Shift Row変換部36およびRound Key加算部37から構成される。

【0006】Round処理は繰り返し行われるが、最終Round処理を含むRound処理の繰り返し回数Nrは、鍵スケジュール部10に入力される鍵長により決められており、表1に示すように規定されている。

【0007】

【表1】

鍵長とラウンド数

鍵長	Nr
128bit	10
192bit	12
256bit	14

【0008】このことから、各鍵長においてそれぞれRound処理の回数をNr-1回行い、最後に最終Round処理を行うものであり、鍵長が128bitではRound処理が9回繰り返され、鍵長が192bitでは11回繰り返され、鍵長が256bitでは13回繰り返され、最後に最終Round処理を行うこととなる。EXOR回路22、Round処理部23および最終Round処理部24には、鍵スケジュール部10で生成されたRound Keyが入力される。

【0009】鍵スケジュール部10は、AESドラフトに示される鍵生成スケジュールに基づいてRound Keyを生成するものであり、そのアルゴリズムを図2に示す。AESブロック暗号の回路化にあたり、AES Proposal仕様 (Rijndael Specification, <http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf>) に2つのハードウェア実装が紹介されている。

【0010】そのうちの1つは、図1に示す全ての機能をそのまま128bit単位でハードウェア化する方法 (以下、従来例1と称す) である。この場合、暗号化と復号

化において、各機能の処理の順番が逆となり、暗号化における処理回路と復号化時における処理回路を別々に用意する必要がある。

【0011】また、表1に示すように鍵長によりRound処理の実施回数を変える必要があるため、鍵長毎に回路を作成する必要がある。さらに、ラウンドファンクション部20で使用するRound Keyは、暗号化時と復号化時において逆順になるため、鍵スケジュール部10において鍵生成の順を暗号化時と復号化時とで逆にする必要がある。したがって、暗号化時と復号化時における鍵スケジュール回路を別々に構成するか、暗号化時と復号化時において鍵スケジュール部10を共通に使用できるような工夫をする必要がある。

【0012】2つ目の方法は、図3に示すように、Byte Sub変換部51とMix Column変換部52を有するコプロセッサ50を作成してByte Sub変換とMix Column変換の機能のみをハードウェア化し、その他の機能をプログラム41に組み込んでソフトウェア化しCPU40で処理する方法 (以下、従来例2と称す) である。

【0013】この場合、処理時間的にCPU40での処理に不向きなByte Sub変換とMix Column変換をコプロセッサ50としてハードウェア化し、他の処理をCPU40に格納されたプログラム41で処理することで回路規模を小さくすることができる。

【0014】

【発明が解決しようとする課題】スマートカード等にAESブロック暗号を組み込むことを想定した場合、暗号回路に要求される性能は、一定水準の処理速度を保ちつつ、回路規模の小型化にある。この要求を前提とした場合、従来提案されている全ての機能を128bit単位でハードウェア化する方法では回路規模が大きすぎ、スマートカードへの搭載が困難となる。また、Byte Sub変換とMix Column変換の機能のみをハードウェア化し、他機能をソフトウェアで処理する方法では、要求処理速度を満足しないといった問題がある。

【0015】また、Round Keyを生成する鍵スケジュール部10では、メモリにすべてのRound Keyを格納すると大きなメモリ容量が必要となり、回路規模が大きくなる。したがって、処理速度を低下させずに回路規模の削減を行うために、メモリにすべての拡大鍵を格納する必要がないような回路構成でRound Keyの生成を行うことが望まれる。

【0016】本発明では、AESブロック暗号を実装する際に、回路規模を小型化するとともに一定水準の高速処理が実現可能な暗号回路の提供を目的とする。

【0017】

【課題を解決するための手段】本発明に係る暗号回路は、暗号鍵から所定の処理ブロック長に対応するビット数の複数のRound Keyを生成し、入力データとRound Keyのうちの1つとEXOR演算を実行するEXOR演算部と、Byte

Sub変換、Shift Row変換、Mix Column変換、Round Key加算を含むRound処理を複数回実行するRound処理部とを備えるラウンドファンクション部により入力データとRound Keyによる暗号化／復号化処理を処理ブロック長毎に実行する暗号回路であって、Round処理部が、入力データを処理ブロック長よりも小さい実行ブロック長に分割する第1セクタと、実行ブロック長毎にRound Keyの値を入力データに加算する第1Round Key加算回路と、第1Round Key加算回路の出力を一時的に格納するとともに処理ブロック長によるShift Row変換を実行する中間レジスタ兼Shift Row変換回路と、中間レジスタ兼Shift Row変換回路の値が実行ブロック長毎に入力されByte Sub変換を実行するByteSub変換回路と、中間レジスタ兼Shift Row変換回路の値が実行ブロック長毎に入力されRound Keyの値を実行ブロック長毎に加算する第2Round Key加算回路と、第2Round Key加算回路の出力に対してMix Column変換を実行するMix Column変換回路と、第1セクタ、中間レジスタ兼Shift Row変換回路、Byte Sub変換回路、Mix Column変換回路の出力のうちいずれか1つを第2Round Key加算回路に出力する第2セクタとを備える。

【0018】ここで、実行ブロック長は8の倍数ビットとすることができ、処理ブロック長が128ビットであり実行ブロック長が32ビットとすることができる。さらに、暗号鍵の鍵長が128ビット、192ビット、256ビットのうちのいずれかとすることができる。

【0019】また、Byte Sub変換回路は、入力データに対して行列演算を実行する復号化用行列演算部と、入力データと復号化用行列演算部の出力とのうちいずれかを出力する第3セクタと、第3セクタから出力されるデータに対して逆数演算を実行する逆数演算部と、逆数演算部から出力されるデータに対して行列演算を実行する暗号化用行列演算部と、逆数演算部の出力と暗号化用行列演算部の出力のうちいずれか一方を出力する第4セクタとを備える構成とすることができる。

【0020】さらに、復号化用行列演算部と暗号化行列演算部は、1クロックで8ビット毎の演算を行うように複数の排他的論理和演算回路を備えている構成とすることができ、1クロックで1ビットの演算を行うように排他的論理和演算回路を備える構成とすることができる。

【0021】また、中間レジスタ兼Shift Row変換回路に入力されるデータのシフト量に関するシフトデータを暗号化時と復号化時において逆順に入力することにより、中間レジスタ兼Shift Row変換回路を暗号化と復号化に共通に使用することができる。

【0022】さらに、Mix Column変換回路は、乗数固定の複数の乗算器と複数の乗算器の排他的論理和を演算するEXOR回路とを備え、各乗算器に入力されるデータと各乗算器に設定された乗数との間で行列演算を実行するように構成できる。この場合、Mix Column変換回路は、8

ビット単位で演算が可能な4つの乗算器と4つの乗算器の出力に基づいて排他的論理和演算を実行するEXOR回路とを有する演算器を4つ備える構成とすることができ、この乗算器は、2つの乗数を制御可能であり暗号化と復号化に共通に使用されるように構成でき、上位ビットからの加算値を制御するように構成することもできる。

【0023】また、暗号鍵を実行ブロック長に応じたビット数に分割して出力する第5セクタと、実行ブロック長毎にデータをラッチするフリップフロップ回路が複数段接続されたシフトレジスタと、シフトレジスタの最終段のフリップフロップ回路の出力と定数群のうちから選択される1つの定数との排他的論理和演算を実行する第1EXOR回路と、シフトレジスタのフリップフロップのうち暗号化時に演算対象となるフリップフロップの出力と復号化時に演算対象となるフリップフロップの出力とが入力されいずれか1つを選択的に出力する第6セクタと、第6セクタの出力をローテーション処理するRot Byte処理回路と、第6セクタの出力とRot Byte処理回路の出力とが入力され、いずれか1つを選択的に出力する第7セクタと、第7セクタの出力に対して実行ブロック長毎のByte Sub変換を実行するSub Byte処理回路と、第6セクタの出力とSub Byte処理回路の出力が入力され、いずれか1つを選択的に出力する第8セクタと、第1EXOR回路の出力と第8セクタの出力とに基づいて排他的論理和演算を実行する第2EXOR回路と、シフトレジスタのフリップフロップのうち暗号化時にその出力が演算対象となるフリップフロップに対して、第2EXOR回路の出力と、隣接する段のフリップフロップの出力とのうちいずれかを選択的に出力するシフトレジスタ部セクタとを備える、暗号鍵から所定の処理ブロック長に対応するビット数の複数のRoundKeyを実行ブロック長に対応するビット数毎に分割された拡大鍵として生成する拡大鍵スケジュール回路を有するように構成できる。

【0024】ここで、シフトレジスタは、32ビット単位でデータ処理を実行する8個のフリップフロップ回路を備え、第6セクタは、フリップフロップのうち下から2段目、4段目、6段目、8段目のフリップフロップの出力が入力され、いずれか1つを出力するように構成できる。

【0025】また、第7セクタに中間レジスタ兼Shift Row変換回路の出力が入力され、Sub Byte処理回路の出力を第2セクタに入力することにより、Sub Byte処理回路とRound処理部のByte Sub変換回路とを共用するように構成できる。

【0026】

【発明の実施の形態】〈ラウンドファンクション部〉AESブロック暗号は、鍵長128／192／256ビット、処理ブロック長128ビットで暗号化／復号化を行

(5)

特開 2003-15522

7

うものであり、図1に示すように、暗号鍵から複数のRound Keyを生成する鍵スケジュール部10と、鍵スケジュール部10から供給されるRound Keyを用いて暗号化／復号化を実行するラウンドファンクション部20で構成される。ラウンドファンクション部20では、排他的論理和演算、Byte Sub変換処理、Shift Row変換処理、Mix Column変換処理、Round Key加算処理などの処理を行う。

【0027】第1実施形態は、このラウンドファンクション部20を実現するための回路を構成するものであり、図4にその回路構成を示す。各回路ブロックではShift Row変換処理を除いて32ビットによる処理を実行し、Shift Row変換処理では128ビットによる処理を実行するものとし、各回路ブロック間のデータの転送は32ビット単位で実施されるものとする。

【0028】このラウンドファンクション部には、入力データを一時格納する入力レジスタ201と、128ビットの入力データから32ビットのデータを選択する第1セクタ202と、第1セクタ202の出力が1つの入力端子に入力される第2セクタ203と、第2セクタ203の出力が入力される第1Round Key加算回路204と、第1Round Key加算回路204に拡大鍵または“0”を加算データとして入力する加算データセクタ205と、第1Round Key加算回路204の出力値を格納するとともに128ビット単位でShift Row変換処理を実行する中間レジスタ兼Shift Row変換回路206と、中間レジスタ兼Shift Row変換回路206の値が32ビット毎に入力されByte Sub変換を実行するByte Sub変換回路207と、中間レジスタ兼Shift Row変換回路206の値が32ビット毎に入力される第2Round Key加算回路208と、第2Round Key加算回路208に拡大鍵または“0”を加算データとして入力する加算データセクタ209と、第2Round Key加算回路208の出力に対してMix Column変換を実行するMix Column変換回路210とを備えている。第2セクタ203には、第1セクタ202、Byte Sub変換回路207、Mix Column変換回路210および中間レジスタ兼Shift Row変換回路206の出力が入力されており、このうちいずれか1つを第2Round Key加算回路204に出力するように構成される。

〈暗号化時の動作スケジュール〉このラウンドファンクション部における暗号化時における動作スケジュールを表2に示す。

【0029】

【表2】

8

### ラウンドファンクション動作スケジュール

Round	Cycle	処理	SEL_B
0	000-003	RoundKey加算	a
1	004-007	ByteSub変換	b
	008	ShiftRow変換	c
	009-012	MixColumn変換 RoundKey加算	c
2	013-016	ByteSub変換	b
	017	ShiftRow変換	c
	018-021	MixColumn変換 RoundKey加算	c
	省略		
Nr-1	#1	ByteSub変換	b
	(Nr-1)*9-1	ShiftRow変換	c
	(Nr-1)*9 - (Nr-1)*9+3	MixColumn変換 RoundKey加算	c
Nr	#2	ByteSub変換	b
	Nr*9-1	ShiftRow変換	d
	Nr*9 - Nr*9+3	RoundKey加算	d

#1:(Nr-1)\*9-5 - (Nr-1)\*9-2

#2:Nr\*9-5 - Nr\*9-2

注:表は暗号化の動作を示している  
復号化ではRound KeyとMixColumnの  
処理順番が入れ代る

【0030】ここで、Round0では、第2セクタ203のセクタ位置をaとして第1Round Key加算回路204により拡大鍵の加算処理を実行する。第1セクタ202により入力レジスタ201内の入力データを32ビット毎に選択して第1RoundKey加算回路204に入力し、鍵スケジュール部から入力されるRound Keyのうち32ビット毎に分割された拡大鍵の加算処理を実行する。入力データおよび拡大鍵を32ビット毎に変更しながら、第1Round Key加算回路204による加算処理を実行することで、第000サイクル〜第003サイクルの4サイクルで128ビットの処理ブロックに対し、図1におけるEXOR処理部22の排他的論理和演算処理を実行することができる。第1Round Key加算回路204による演算結果は、32ビット毎に中間レジスタ兼Shift Row変換回路206に順次格納される。

【0031】Round1では、図1におけるRound処理23を実行するものであり、Byte Sub変換処理31、Shift Row変換処理32、Mix Column変換処理33、Round Key加算処理34を実行する。このために、まず、第004サイクル〜第007サイクルで、第2セクタ203のセクタ位置をbとし、中間レジスタ兼Shift Row変換回路206に格納されているデータを32ビットずつシフトさせながら読み出してByte Sub変換回路207に入力する。このとき、加算データセクタ205により選択さ

れるデータを“0”とすることで、第1 Round Key加算回路204をマスク状態とする。Byte Sub変換回路207による演算結果は、32ビット毎に中間レジスタ兼Shift Row変換回路206に順次格納される。このことにより、128ビットのByte Sub変換処理が行われ、演算結果が中間レジスタ兼Shift Row変換回路206に格納されることとなる。

【0032】次に、第008サイクルにおいて、Shift Row変換処理を実行する。中間レジスタ兼Shift Row変換回路206では、128ビット単位でShift Row変換処理を行うことが可能となっており、この第008サイクルにおいて、128ビットのShift Row変換処理を実行する。このとき、第2セクタ203のセクタ位置はいずれの位置であってもよいが、次のサイクルでの処理を考慮してcの位置とすることが好ましい。

【0033】第009サイクル～第012サイクルでは、Mix Column変換処理およびRound Key加算処理を実行する。ここでは、中間レジスタ兼Shift Row変換回路206に格納されているデータを32ビットずつシフトさせながら読み出して第2 Round Key加算回路208に入力する。このとき、加算データセクタ209により選択されるデータを“0”とすることで、第2 Round Key加算回路208をマスク状態とする。また、第2セクタ203のセクタ位置をcとすることにより、Mix Column変換回路210でMix Column変換処理が実行されたデータが第2セクタ203を介して第1 Round Key加算回路204に入力される。加算データセクタ205により選択されるデータは、鍵スケジュール部から入力される拡大鍵が選択されており、第1 Round Key加算回路204においてRound Key加算処理が実行される。Mix Column変換回路210でMix Column変換処理され、第1 Round Key加算回路204においてRound Key加算処理された演算結果は、32ビット毎に中間レジスタ兼Shift Row変換回路206にそれぞれシフトされながら格納される。このことにより、第009サイクル～第012サイクルで、Mix Column変換処理およびRound Key加算処理を実行した128ビットの演算結果が中間レジスタ兼Shift Row変換回路206に格納される。このようにして、第004サイクル～第012サイクルの9サイクルで1つのRound処理が実行される。

【0034】以下、第2 Round～第(Nr-1) Roundにおいて、第1 Roundと同様の処理を実行する（ただし、Nrは最終Round処理を含むRound処理の繰り返し回数であり、表1に示すように、鍵長により異なる回数が規定されている）。

【0035】第(Nr) Round（最終Round）では、図1における最終Round処理24を実行するものであり、Byte Sub変換処理35、Shift Row変換処理36、Round Key加算処理37を実行する。

【0036】このために、第(Nr\*9-5)サイクル～

第(Nr\*9-2)サイクルにおいて、第2セクタ203のセクタ位置をbとし、中間レジスタ兼Shift Row変換回路206に格納されているデータを32ビットずつシフトさせながら読み出してByte Sub変換回路207に入力する。このとき、加算データセクタ205により選択されるデータを“0”とすることで、第1 Round Key加算回路204をマスク状態とする。Byte Sub変換回路207による演算結果は、32ビット毎に中間レジスタ兼Shift Row変換回路206に順次格納される。このことにより、128ビットのByte Sub変換処理が行われ、演算結果が中間レジスタ兼Shift Row変換回路206に格納されることとなる。

【0037】次に、第(Nr\*9-1)サイクルにおいて、128ビットのShift Row変換処理を実行する。このとき、第2セクタ203のセクタ位置はいずれの位置であってもよいが、次のサイクルでの処理を考慮してdの位置とすることが好ましい。

【0038】第(Nr\*9)サイクル～第(Nr\*9+3)サイクルでは、Round Key加算処理を実行する。ここでは、第2セクタ204のセクタ位置をdとすることで、中間レジスタ兼Shift Row変換回路206に格納されているデータを32ビットずつシフトさせながら読み出して第2セクタ203を介して第1 Round Key加算回路204に入力する。このとき、加算データセクタ205により選択されるデータを鍵スケジュール部から供給される拡大鍵とすることで、第1 Round Key加算回路204による32ビット毎のRound Key加算処理を実行することができる。第1 Round Key加算回路204においてRound Key加算処理された演算結果は、32ビット毎に中間レジスタ兼Shift Row変換回路206にそれぞれシフトされながら格納される。このことにより、第(Nr\*9)サイクル～第(Nr\*9+3)サイクルで、Round Key加算処理を実行した128ビットの演算結果が中間レジスタ兼Shift Row変換回路206に格納される。このようにして、第(Nr\*9-5)サイクル～第(Nr\*9+3)サイクルの9サイクルで最終Round処理が実行される。

〈復号化時の動作スケジュール〉このラウンドファンクション部における復号化時の動作は暗号化時の逆順で処理が実行される。このときの動作スケジュールを表3に示す。

【0039】

【表3】

## ラウンドファンクション動作スケジュール

Round	Cycle	処理	SEL B
0	000-003	RoundKey加算	a
1	004	ShiftRow変換	b
	005-008	ByteSub変換	b
	009-012	RoundKey加算 MixColumn変換	c
2	013	ShiftRow変換	b
	014-017	ByteSub変換	b
	018-021	RoundKey加算 MixColumn変換	a
	省略		
Nr-1	(Nr-1)*9-5	ShiftRow変換	b
	#1	ByteSub変換	b
	(Nr-1)*9 - (Nr-1)*9+3	RoundKey加算 MixColumn変換	a
Nr	Nr*9-5	ShiftRow変換	b
	#2	ByteSub変換	b
	Nr*9 - Nr*9+3	RoundKey加算	d

#1:(Nr-1)\*9-4-(Nr-1)\*9-1

#2:Nr\*9-4-Nr\*9-1

【0040】ここで、Round 0では、第2セクタ203のセクタ位置をaとして第1Round Key加算回路204により拡大鍵の加算処理を実行する。第1セクタ202により入力レジスタ201内の入力データを32ビット毎に選択して第1RoundKey加算回路204に入力し、鍵スケジュール部から入力されるRound Keyのうち32ビット毎に分割された拡大鍵の加算処理を実行する。このとき、第1セクタ202を介して入力されるデータは暗号化時の入力順と逆に入力され、かつ鍵スケジュール部から入力される拡大鍵の入力順も暗号化時の逆に入力することとする。このように、入力データおよび拡大鍵を32ビット毎に変更しながら、第1Round Key加算回路204による加算処理を実行することで、第000サイクル～第003サイクルの4サイクルで128ビットの処理ブロックに対するRound Key加算処理を実行することができる。第1Round Key加算回路204による演算結果は、32ビット毎に中間レジスタ兼Shift Row変換回路206に順次格納される。

【0041】Round 1では、Shift Row変換、Byte Sub変換、Round Key加算、Mix Column変換の順に処理を実行する。このために、まず、第004サイクルにおいて、中間レジスタ兼Shift Row変換回路206において、128ビット単位でShift Row変換処理を実行する。この場合、暗号化時におけるShift Row変換処理と同じ処理が行われる。また、第2セクタ203のセクタ位置はいずれの位置であってもよいが、次のサイクルでの処理を考慮してbの位置とすることが好ましい。

【0042】次に、第005サイクル～第008サイクルで、第2セクタ203のセクタ位置をbとし、中間レジスタ兼Shift Row変換回路206に格納されているデータを32ビットずつシフトさせながら読み出してByte Sub変換回路207に入力する。このとき、加算データセクタ205により選択されるデータを“0”とすること、第1Round Key加算回路204をマスク状態とする。Byte Sub変換回路207による演算結果は、32ビット毎に中間レジスタ兼Shift Row変換回路206に順次格納される。この場合も、暗号化時の逆変換処理となるようにByte Sub変換処理を実行するものであり、詳細は後述する。このことにより、128ビットのByte Sub変換処理が行われ、演算結果が中間レジスタ兼Shift Row変換回路206に格納されることとなる。

【0043】第009サイクル～第012サイクルでは、Round Key加算処理およびMix Column変換処理を実行する。ここでは、中間レジスタ兼Shift Row変換回路206に格納されているデータを32ビットずつシフトさせながら読み出して第2Round Key加算回路208に入力する。このとき、加算データセクタ209により選択されるデータを鍵スケジュール部から入力される拡大鍵とする。また、第2セクタ203のセクタ位置をcとして、Mix Column変換回路210の出力を第2セクタ203を介して第1Round Key加算回路204に入力する。このとき、加算データセクタ205により選択されるデータを“0”とすることにより、第1Round Key加算回路204をマスク状態とする。この場合も、暗号化時と逆変換処理となるようにMix Column変換処理を実行するものであり、詳細は後述する。このことにより、第2Round Key加算回路208によるRound Key加算処理とMix Column変換回路210によるMix Column変換処理を実行した128ビットの演算結果が中間レジスタ兼Shift Row変換回路206に格納される。このようにして、第004サイクル～第012サイクルの9サイクルで1つのRound処理が実行される。

【0044】以下、第2Round～第(Nr-1)Roundにおいて、第1Roundと同様の処理を実行する（ただし、Nrは最終Round処理を含むRound処理の繰り返し回数であり、表1に示すように、鍵長により異なる回数が規定されている）。

【0045】第(Nr)Round（最終Round）では、Shift Row変換処理、Byte Sub変換処理、Round Key加算処理を実行する。このために、第(Nr\*9-5)サイクルにおいて、128ビットのShift Row変換処理を実行する。このとき、第2セクタ203のセクタ位置はいずれの位置であってもよいが、次のサイクルでの処理を考慮してbの位置とすることが好ましい。

【0046】次に、第(Nr\*9-4)サイクル～第(Nr\*9-1)サイクルにおいて、第2セクタ203のセクタ位置をbとし、中間レジスタ兼Shift Row変換回



路 206 に格納されているデータを 32 ビットずつシフトさせながら読み出して Byte Sub 変換回路 207 に入力する。このとき、加算データセクタ 205 により選択されるデータを "0" とすることで、第 1 Round Key 加算回路 204 をマスク状態とする。Byte Sub 変換回路 207 による演算結果は、32 ビット毎に中間レジスタ兼 Shift Row 変換回路 206 に順次格納される。このことにより、128 ビットの Byte Sub 変換処理が行われ、演算結果が中間レジスタ兼 Shift Row 変換回路 206 に格納されることとなる。

【0047】第  $(Nr * 9)$  サイクル～第  $(Nr * 9 + 3)$  サイクルでは、Round Key 加算処理を実行する。ここでは、第 2 セクタ 204 のセクタ位置を  $d$  とすることで、中間レジスタ兼 Shift Row 変換回路 206 に格納されているデータを 32 ビットずつシフトさせながら読み出して第 2 セクタ 203 を介して第 1 Round Key 加算回路 204 に入力する。このとき、加算データセクタ 205 により選択されるデータを鍵スケジュール部から供給される拡大鍵とすることで、第 1 Round Key 加算回路 204 による 32 ビット毎の Round Key 加算処理を実行することができる。第 1 Round Key 加算回路 204 において Round Key 加算処理された演算結果は、32 ビット毎に中間レジスタ兼 Shift Row 変換回路 206 にそれぞれシフトされながら格納される。このことにより、第  $(Nr * 9)$  サイクル～第  $(Nr * 9 + 3)$  サイクルで、Round Key 加算処理を実行した 128 ビットの演算結果が中間レジスタ兼 Shift Row 変換回路 206 に格納される。このようにして、第  $(Nr * 9 - 5)$  サイクル～第  $(Nr * 9 + 3)$  サイクルの 9 サイクルで最終 Round 処理が実行される。

〈中間値レジスタ兼 Shift Row 変換回路〉中間値レジスタ兼 Shift Row 変換回路の 1 実施例を図 5 に示す。

【0048】ここでは、8 ビット単位で処理を行うシフトレジスタを 4 個設けた構成となっている。1 番目のシフトレジスタは、4 つのフリップフロップ 302, 304, 306, 308 が順次接続される構成でなり、各フリップフロップ 302, 304, 306, 308 にはそれぞれ入力を選択するセクタ 301, 303, 305, 307 が接続されている。1 番目のセクタ 301 には、入力データ  $IN_0$  とフリップフロップ 302 の出力とが入力されておりいずれか一方をフリップフロップ 302 に入力するように構成されている。また、2～4 番目のセクタ 303, 305, 307 にはそれぞれ前段のフリップフロップ 302, 304, 306 の出力およびフリップフロップ 304, 306, 308 の出力とが入力されておりいずれか一方をフリップフロップ 304, 306, 308 に入力するように構成されている。

【0049】2 番目のシフトレジスタは、4 つのフリップフロップ 312, 314, 316, 318 が順次接続される構成でなり、各フリップフロップ 312, 31

4, 316, 318 にはそれぞれ入力を選択するセクタ 311, 313, 315, 317 が接続されている。1 番目のセクタ 311 には、入力データ  $IN_1$  とフリップフロップ 312 の出力およびフリップフロップ 318 の出力が入力されておりいずれか一方をフリップフロップ 312 に入力するように構成されている。また、2～4 番目のセクタ 313, 315, 317 にはそれぞれ前段のフリップフロップ 312, 314, 316 の出力およびフリップフロップ 314, 316, 318 の出力とが入力されておりいずれか一方をフリップフロップ 314, 316, 318 に入力するように構成されている。

【0050】3 番目のシフトレジスタは、4 つのフリップフロップ 322, 324, 326, 328 が順次接続される構成でなり、各フリップフロップ 322, 324, 326, 328 にはそれぞれ入力を選択するセクタ 321, 323, 325, 327 が接続されている。1 番目のセクタ 321 には、入力データ  $IN_2$  とフリップフロップ 322 の出力およびフリップフロップ 326 の出力が入力されておりいずれか一方をフリップフロップ 322 に入力するように構成されている。また、2 番目のセクタ 323 には前段のフリップフロップ 322 の出力、フリップフロップ 324 の出力およびフリップフロップ 328 の出力が入力されておりいずれか一方をフリップフロップ 324 に入力するように構成されている。3 番目のセクタ 325 には前段のフリップフロップ 324 の出力、フリップフロップ 326 の出力およびフリップフロップ 322 の出力が入力されておりいずれか一方をフリップフロップ 326 に入力するように構成されている。4 番目のセクタ 327 には前段のフリップフロップ 326 の出力、フリップフロップ 328 の出力およびフリップフロップ 324 の出力が入力されておりいずれか一方をフリップフロップ 328 に入力するように構成されている。

【0051】4 番目のシフトレジスタは、4 つのフリップフロップ 332, 334, 336, 338 が順次接続される構成でなり、各フリップフロップ 332, 334, 336, 338 にはそれぞれ入力を選択するセクタ 331, 333, 335, 337 が接続されている。1 番目のセクタ 331 には、入力データ  $IN_3$  とフリップフロップ 332 の出力およびフリップフロップ 334 の出力が入力されておりいずれか一方をフリップフロップ 332 に入力するように構成されている。また、2 番目のセクタ 333 には前段のフリップフロップ 332 の出力、フリップフロップ 334 の出力およびフリップフロップ 336 の出力が入力されておりいずれか一方をフリップフロップ 334 に入力するように構成されている。3 番目のセクタ 335 には前段のフリップフロップ 334 の出力、フリップフロップ 336 の出力およびフリップフロップ 338 の出力が入力されておりいずれ



(9)

特開2003-15522

15

か一方をフリップフロップ336に入力するように構成されている。4番目のセクタ337には前段のフリップフロップ336の出力、フリップフロップ338の出力およびフリップフロップ332の出力が入力されておりいずれか一方をフリップフロップ338に入力するように構成されている。

【0052】このように構成した中間値レジスタ兼Shift Row変換回路を、各処理における中間値レジスタとして動作させる場合には、入力データIN0～IN3に各8ビット単位のデータを入力することで、各サイクルにおいて32ビット単位で処理されたデータを格納することができる。さらに、各セクタ301～337のセクタ位置をbにして、各フリップフロップのデータを後段にシフトさせつつ、入力データIN0～IN3にそれぞれ8ビット毎のデータを入力することで、4サイクルで128ビット

## データ配列と処理順番

	Row →			
Column ↑	a00	a01	a02	a03
	a10	a11	a12	a13
	a20	a21	a22	a23
	a30	a31	a32	a33

暗号化

	← Row			
Column ↑	a00	a01	a02	a03
	a10	a11	a12	a13
	a20	a21	a22	a23
	a30	a31	a32	a33

復号化

【0055】表4左に示すように、各Row（行）におけるデータが左端のColumnから順に配列されているような場合には、暗号化時には、左端のColumnから順に処理が実行される。また、復号化時には、表4右に示すように、右端のColumnから順に処理が実行される。

【0056】Shift Row変換処理では、暗号化時には、表4左に示されているような配列のデータ配列をRow毎に異なるシフト量でローテートシフトする。具体的に

## [暗号化]

処理前			
a00	a01	a02	a03
a10	a11	a12	a13
a20	a21	a22	a23
a30	a31	a32	a33

左1Byteローテートシフト

左2Byteローテートシフト

左3Byteローテートシフト

処理後			
a00	a01	a02	a03
a11	a12	a13	a10
a22	a23	a20	a21
a33	a30	a31	a32

【0058】また、復号化時には、暗号化時と逆変換処理となるように、表4左に示されているような配列のデータ配列からRow毎に異なるシフト量でローテートシフトする。具体的には、表6に示すように、1行目をそのままとし、2行目を左3Byteローテートシフト、3行目

## [復号化]

処理前			
a00	a01	a02	a03
a10	a11	a12	a13
a20	a21	a22	a23
a30	a31	a32	a33

左3Byteローテートシフト

左2Byteローテートシフト

左1Byteローテートシフト

処理後			
a00	a01	a02	a03
a13	a10	a11	a12
a22	a23	a20	a21
a31	a32	a33	a30

トのデータを入力することができる。128ビットのデータ入力が完了した場合、1サイクル目に入力された4つの8ビットデータは、それぞれフリップフロップ308、318、328、338にラッチされていることとなる。

【0053】次にShift Row変換の動作について説明する。Rijndaelアルゴリズムでは、入力データをそれぞれ8bitのデータa00～a33に分割しこれをマトリクスとして処理を行い、暗号化と復号化でシフトの向きが逆になっている。本発明では、データの処理順をColumn（列）配列順とし、暗号化時と復号化時において逆順に処理することにより、同じ処理でShift Row変換処理を実現することができる。

【0054】

【表4】

は、表5に示すように、1行目をそのままとし、2行目を左1Byteローテートシフト、3行目を左2Byteローテートシフト、4行目を左3Byteローテートシフトする。このことにより、表5左に示す処理前の状態から表5右の処理後の状態となる。

【0057】

【表5】

を左2Byteローテートシフト、4行目を左1Byteローテートシフトする。このことにより、表6左に示す処理前の状態から表6右の処理後の状態となる。

【0059】

【表6】

(10)

特開 2 0 0 3 - 1 5 5 2 2

17

【0060】本実施形態では、図5に示すような中間値レジスタ兼Shift Row変換回路を用いている。このことから、128ビットのデータの入力完了した段階で、最初のサイクルで入力されたデータが最後段のフリップフロップ308、318、328、338にラッチされており、前段のフリップフロップに順にデータがラッチされている。データを取り出す場合には、1サイクル毎

【暗号化】

処理前			
a03	a02	a01	a00
a13	a12	a11	a10
a23	a22	a21	a20
a33	a32	a31	a30

右1Byteローテートシフト  
右2Byteローテートシフト  
右3Byteローテートシフト

【0062】表5と同様のローテートシフトを行うためには、表7の右に示すように、1行目をそのままとし、2行目を右1Byteローテートシフト、3行目を右2Byteローテートシフト、4行目を右3Byteローテートシフトすることとなる。

【0063】このような暗号化時におけるShift Row変換処理を行うためには、図5に示す中間値レジスタ兼Shift Row変換回路を用いて各セクタを切替制御し、128ビット単位で一括してデータの入れ換えを行う。

【0064】1番目のRowについては、シフトの必要がないため、各セクタ301、303、305、307のセクタ位置をaとする。2番目のRowについては、右1Byteローテートシフトを実行するために、セクタ311のセクタ位置をcとし、それ以外のセクタ313、315、317のセクタ位置をbとする。3番目のRowについては、右2Byteローテートシフトを実行するために、各セクタ321、323、325、327のセクタ位置をcとする。4番目のRowについては、右に3byteローテートシフトを実行するために、各セクタ331、333、335、337のセクタ位置をcとする。

【復号化】

処理前			
a00	a01	a02	a03
a10	a11	a12	a13
a20	a21	a22	a23
a30	a31	a32	a33

右1Byteローテートシフト  
右2Byteローテートシフト  
右3Byteローテートシフト

【0069】表6と同様のローテートシフトを行うためには、表9の右に示すように、1行目をそのままとし、2行目を右1Byteローテートシフト、3行目を右2Byteローテートシフト、4行目を右3Byteローテートシフトすることとなる。

【0070】したがって、前述した暗号化時におけるShift Row変換時と同様に、中間値レジスタ兼Shift Row変換回路の各セクタのセクタ位置を設定して、表8に示すような暗号化時におけるローテートシフト処理と全

18

に右側に1Byteずつシフトしながら右端に位置する最後段のフリップフロップからデータが取り出されることとなる。したがって、データの処理順が右端側からであることを考慮してデータを配列し直すと、暗号化時のShift Row処理前は表7の左のような形態となる。

【0061】

【表7】

処理後			
a03	a02	a01	a00
a10	a13	a12	a11
a21	a20	a23	a22
a32	a31	a30	a33

置をcとする。

【0065】前述したようなShift Row変換処理を実行する前における中間値レジスタ兼Shift Row変換回路の各フリップフロップがラッチしている出力データを図5に示すようにそれぞれb00～b33とすると、各出力データは表8の右に示すような配列で各フリップフロップの出力にラッチされることとなる。

【0066】

【表8】

ShiftRow変換動作モデル

ShiftRow変換前			
b03	b02	b01	b00
b13	b12	b11	b10
b23	b22	b21	b20
b33	b32	b31	b30

→

ShiftRow変換後			
b03	b02	b01	b00
b10	b13	b12	b11
b21	b20	b23	b22
b32	b31	b30	b33

【0067】復号化時には、表4のように右側の列から処理が実行されるため、表9左に示すように、データが配列される。

【0068】

【表9】

処理後			
a00	a01	a02	a03
a13	a10	a11	a12
a22	a23	a20	a21
a31	a32	a33	a30

く同じ処理を実行することにより、復号化時におけるShift Row変換処理を実行することが可能となる。

【0071】このことにより、同一の中間値レジスタ兼Shift Row変換回路で暗号化時および復号化時におけるShift Row変換処理を共用することができる。

〈Mix Column変換回路〉この実施形態で採用されるMix Column変換回路の一例を図6に示す。

【0072】このMix Column変換回路は4つの第1演算器351、第2演算器352、第3演算器353、第4

演算器 354 を含んでいる。第 1 演算器 351 は、それぞれ 8 ビット単位で演算処理を実行する第 1 乗算器 361、第 2 乗算器 362、第 3 乗算器 363、第 4 乗算器 364 と、各乗算器 361～364 の出力の排他的論理和演算を実行する EXOR 回路 365 とで構成されている。その他の第 2 演算器 352～第 4 演算器 354 についても、図示していないが、第 1 乗算器～第 4 乗算器の 4 つの乗算器と EXOR 回路が設けられている。

【0073】(a0j, a1j, a2j, a3j) で構成される第 j 番目の Column を (b0j, b1j, b2j, b3j) で構成される Column に変換処理を行うものとした場合、変換処理後の第 j 番目の Column のデータ (b0j, b1j, b2j, b3j) は、次のように表すことができる。

暗号化

$$b0j = 02*a0j + 03*a1j + 01*a2j + 01*a3j$$

$$b1j = 01*a0j + 02*a1j + 03*a2j + 01*a3j$$

$$b2j = 01*a0j + 01*a1j + 02*a2j + 03*a3j$$

$$b3j = 03*a0j + 01*a1j + 01*a2j + 02*a3j$$

復号化

$$b0j = 0E*a0j + 0B*a1j + 0D*a2j + 09*a3j$$

$$b1j = 09*a0j + 0E*a1j + 0B*a2j + 0D*a3j$$

$$b2j = 0D*a0j + 09*a1j + 0E*a2j + 0B*a3j$$

$$b3j = 0B*a0j + 0D*a1j + 09*a2j + 0E*a3j$$

ここで、各項に乗算されている係数はそれぞれ 2 桁の 16 進数である。

【0074】このような Mix Column 変換処理を実行するために、Column 単位の 32 ビットデータをそれぞれ第 1 演算器 351～第 4 演算器 354 に入力して、第 1 乗算器～第 4 乗算器による乗算および EXOR 回路による演算を行う。

【0075】各演算器 351～354 に設けられる乗算器 361～364 は、暗号化および復号のいずれの場合にも共通に使用することができるように、暗号化時の係数および復号化時の係数を備えており、演算時に選択することが可能に構成されている。

【0076】第 1 演算器 351 の第 1 乗算器 361 では、入力されるデータに対して 0x02 倍または 0x0E 倍のいずれかの乗算が可能となっている。第 2 乗算器 362 では、入力されるデータに対して 0x03 倍または 0x0B 倍のいずれかの乗算が可能となっている。第 3 乗算器 363 では、入力されるデータに対して 0x01 倍または 0x0D 倍のいずれかの乗算が可能となっている。第 4 乗算器 364 では、入力されるデータに対して 0x01 倍または 0x09 倍のいずれかの乗算が可能となっている。

【0077】第 2 演算器 352 の第 1 乗算器では、入力されるデータに対して 0x01 倍または 0x09 倍のいずれかの乗算が可能となっている。第 2 乗算器では、入力されるデータに対して 0x02 倍または 0x0E 倍のいずれかの乗算が可能となっている。第 3 乗算器では、入力されるデータに対して 0x03 倍または 0x0B 倍のいずれかの乗算が可能と

なっている。第 4 乗算器では、入力されるデータに対して 0x01 倍または 0x0D 倍のいずれかの乗算が可能となっている。

【0078】第 3 演算器 353 の第 1 乗算器では、入力されるデータに対して 0x01 倍または 0x0D 倍のいずれかの乗算が可能となっている。第 2 乗算器では、入力されるデータに対して 0x01 倍または 0x09 倍のいずれかの乗算が可能となっている。第 3 乗算器では、入力されるデータに対して 0x02 倍または 0x0E 倍のいずれかの乗算が可能となっている。第 4 乗算器では、入力されるデータに対して 0x03 倍または 0x0B 倍のいずれかの乗算が可能となっている。

【0079】第 4 演算器 354 の第 1 乗算器では、入力されるデータに対して 0x03 倍または 0x0B 倍のいずれかの乗算が可能となっている。第 2 乗算器では、入力されるデータに対して 0x01 倍または 0x0D 倍のいずれかの乗算が可能となっている。第 3 乗算器では、入力されるデータに対して 0x01 倍または 0x09 倍のいずれかの乗算が可能となっている。第 4 乗算器では、入力されるデータに対して 0x02 倍または 0x0E 倍のいずれかの乗算が可能となっている。

【0080】このような第 1 演算器 351～第 4 演算器 354 の第 1 乗算器～第 4 乗算器の各係数を暗号化時および復号化時において変更することにより、暗号化時と復号化時における回路構成を共通に使用することが可能となる。

〈Mix Column 変換回路の乗算器〉 Mix Column 変換回路中に含まれる乗算器の一例を図 7 に示す。

【0081】入力される 8 ビットデータ (a7, a6, a5, a4, a3, a2, a1, a0) に対し、係数 (b3, b2, b1, b0) の乗算を行うものとする。この場合、8 ビットデータ (a7, a6, a5, a4, a3, a2, a1, a0) に係数の各ビットを乗算する部分積演算部 375～378 を設けている。さらに、係数の最上位ビットを乗算する部分積演算部 375 の演算結果に対して、部分積演算部 376 の演算結果を 1 ビットシフトさせて加算する加算器 371、部分積演算部 377 の演算結果をさらに 1 ビットシフトさせて加算する加算器 372、部分積演算部 378 の演算結果をさらに 1 ビットシフトさせて加算する加算器 373 を備えている。加算器 373 の演算結果と部分積演算部 375、各加算器 371～373 でオーバーフローしたキャリーが入力され、除数により除算される除算器 374 が設けられている。

【0082】このような構成により、係数 (b3, b2, b1, b0) として、暗号化時における係数と復号化時における係数とを選択的にセットすることで Mix Column 変換処理を暗号化時と復号化時において共通に使用することができる。

【0083】(b3, b2, b1, b0) としてセットされる係数は、前述したように、各乗算器について 2 つずつ設定さ

れている。各乗算器における係数の組み合わせは、(0x02, 0x0E), (0x03, 0x0B), (0x01, 0x0D), (0x01, 0x09) の4通りであり、これを下位4ビットで表示すると、(0010, 1110), (0011, 1011), (0001, 1101), (0001, 1001) となる。このような係数において共通するビットの演算は部分積の制御を行わず、異なるビットの演算は加算処理を制御することで、回路規模を小さくすることが可能である。

【0084】たとえば、係数が(0x01, 0x0D)の組み合わせの場合、2進数で表現すると(0001, 1101)となり、上位2ビットの部分積の加算結果を下位2ビットの部分積に加算するか否かを制御することで、2つの係数を選択して乗算することが可能となる。係数(0x01, 0x0D)の組み合わせの場合における回路構成の例を図8に示す。

【0085】図8では、入力される8ビットデータ(a7, a6, a5, a4, a3, a2, a1, a0)に対し、1ビットシフトさせて加算処理を実行する第1加算器381が設けられている。第1加算器381の出力は、制御ロジック382を介して第2加算器383に入力されている。この第2加算器383は、係数の最下位ビットによる部分積演算結果を加算するものであって、入力される8ビットデータに対して3ビットシフトさせて加算処理を実行するように構成されている。

【0086】加算器383の演算結果と第1加算器381および第2加算器383でオーバーフローしたキャリーが入力され、除数により除算される除算器384が設けられている。

【0087】制御ロジック382は、係数が0x01である場合には、上位2ビットの演算結果である第1加算器381の出力を第2加算器383に出力しない。また、制御ロジック382は、係数が0x0Dである場合には、上位2ビットの演算結果である第1加算器381の出力を第2加算器383に出力するように構成される。

【0088】ここでの乗算は既約多項式 $M(x) = x^8 + x^4 + x^3 + x + 1$ とするGF(2<sup>8</sup>)上での乗算であり、加算はGF(2)上での加算であるため、排他的論理和演算で実現することが可能である。

【0089】このように、2つの係数の異なるビットにおける部分積の加算処理を制御することにより回路規模を小さくすることができ、回路規模を削減することが可能となる。

〈鍵スケジュール部〉鍵スケジュール部の回路構成の一例を図9に示す。

【0090】この鍵スケジュール部は、主に、拡大鍵生成ロジック部101、拡大鍵レジスタ120および鍵入力レジスタ131とから構成されている。鍵入力レジスタ131は、8個の32ビットレジスタk0~k7から構成される256ビットのレジスタであり、レジスタk0から順に暗号鍵が32ビット毎に格納される。暗号鍵が25

6ビットの場合にはレジスタk0~k7全てにデータが格納され、暗号鍵が192ビットの場合にはレジスタk0~k5にデータが格納され、暗号鍵が128ビットの場合にはレジスタk0~k3にデータが格納される。

【0091】鍵入力レジスタ131には、各レジスタk0~k7のうちいずれか1つの値を選択的に出力するセクタ132が接続されている。このセクタ132は、鍵入力レジスタ131の256ビットデータから32ビットのデータを選択して拡大鍵レジスタ120の最下位に

10

入力するものである。

【0092】拡大鍵レジスタ120は、32ビット単位での処理が可能な8個のフリップフロップ121~128が順次接続されたシフトレジスタで構成されている。最下位に位置するフリップフロップ128には、セクタ132の出力と拡大鍵生成ロジック部101の出力を選択するセクタ113の出力が入力される。フリップフロップ128の出力W7KEYは、フリップフロップ127に入力されている。フリップフロップ127の出力W6KEYは、フリップフロップ126の前段にあるセクタ112に入力されている。セクタ112には、フリップフロップ127の出力W6KEYと拡大鍵生成ロジック101の出力とが入力されており、いずれか一方をフリップフロップ126に入力するように構成されている。

20

【0093】フリップフロップ126の出力W5KEYは、フリップフロップ125に入力されている。フリップフロップ125の出力W4KEYは、フリップフロップ124の前段にあるセクタ111に入力されている。セクタ111には、フリップフロップ125の出力W4KEYと拡大鍵生成ロジック101の出力とが入力されており、いずれか一方をフリップフロップ124に入力するように構成されている。

30

【0094】フリップフロップ124の出力W3KEYは、フリップフロップ123に入力されている。フリップフロップ123の出力W2KEYは、フリップフロップ122に入力されている。フリップフロップ122の出力W1KEYは、フリップフロップ121に入力されている。

【0095】拡大鍵生成ロジック部101は、拡大鍵生成用定数Rconが格納されているROM102、ROM102から読み出される値と読出信号RCON#ENとの論理積演算を行うAND回路103、拡大鍵レジスタ120の最上位に位置するフリップフロップ121の出力W0KEYとAND回路103の出力が入力されて排他的論理和演算を実行するEXOR回路104を含んでいる。

【0096】また、拡大鍵生成ロジック部101は、フリップフロップ121の出力W0KEY、フリップフロップ124の出力W3KEY、フリップフロップ126の出力W5KEY、フリップフロップ128の出力W7KEYが入力され、いずれか1つを選択的に出力するセクタ105を備えている。セクタ105の出力は、データのローテーション処理を行うRot Byte回路106、セクタ107、

50

セクタ109に入力されている。セクタ107には、Rot Byte回路106の出力とセクタ105の出力が入力されており、いずれかをSub Byte回路108に供給する。Sub Byte回路108は、32ビット分のByte Sub変換処理を実行するものであり、その出力がセクタ109に供給する。セクタ109は、Sub Byte回路108の出力とセクタ105の出力が入力され、いずれか一方を出力する。拡大鍵生成ロジック部101は、EXOR回路104の出力とセクタ109の出力が入力され、排他的論理和演算を実行するEXOR回路110を備えている。

【0097】このように構成される鍵スケジュール部では、ラウンドファンクション部のRound Key加算処理に用いる拡大鍵の生成、暗号化時における鍵入力レジスタの書き換えと、暗号化と復号化終了に伴う拡大鍵初期値のセットアップ、復号化時の鍵入力レジスタの書き換えに伴う拡大鍵初期値のセットアップなどの機能を

拡大鍵スケジュール(例:鍵長=256bit)

No.	暗号化	復号化	
00	W00=W00	W59	Initial Round Key
01	W01=W01	W58	
02	W02=W02	W57	
03	W03=W03	W56	
04	W04=W04	W55	
05	W05=W05	W54	Round Key01
06	W06=W06	W53	
07	W07=W07	W52	Round Key02
08	W08=W08 SubByte(RotByte(W07)) Rcon[1]	W51=W59 W58	
09	W09=W01 W08	W50=W58 W57	
10	W10=W02 W09	W49=W57 W56	
11	W11=W03 W10	W48=W56 SubByte(RotByte(W55)) Rcon[7]	
12	W12=W04 SubByte(W11)	W47=W55 W54	Round Key03
13	W13=W05 W12	W46=W54 W53	
14	W14=W06 W13	W45=W53 W52	
15	W15=W07 W14	W44=W52 SubByte(W51)	
16	W16=W08 SubByte(RotByte(W15)) Rcon[2]	W43=W51 W50	
17	W17=W09 W16	W42=W50 W49	Round Key04
18	W18=W10 W17	W41=W49 W48	
19	W19=W11 W18	W40=W48 SubByte(RotByte(W47)) Rcon[6]	
20	W20=W12 SubByte(W19)	W39=W47 W46	
21	W21=W13 W20	W38=W46 W45	
22	W22=W14 W21	W37=W45 W44	Round Key05
23	W23=W15 W22	W36=W44 SubByte(W43)	
	省略		
52	W52=W44 SubByte(W51)	W07=W15 W14	Round Key13
53	W53=W45 W52	W06=W14 W13	
54	W54=W46 W53	W05=W13 W12	
55	W55=W47 W54	W04=W12 SubByte(W11)	
56	W56=W48 SubByte(RotByte(W55)) Rcon[7]	W03=W11 W10	
57	W57=W49 W56	W02=W10 W09	Round Key14
58	W58=W50 W57	W01=W09 W08	
59	W59=W51 W58	W00=W08 SubByte(RotByte(W07)) Rcon[1]	

【0100】ここで、暗号化時の拡大鍵W08は、 $W08=W00 \text{ SubByte(Rot Byte(W07)) Rcon[1]}$ の式により、W00とSub Byte(Rot Byte(W07))と定数Rcon[1]との排他的論理和で演算される。 $A \wedge A=A$ という性質を利用すると、拡大鍵W00は、 $W00=W08 \text{ SubByte(Rot Byte(W07)) Rcon[1]}$ で表され、W08とW07から生成できることがわかる。よって、復号化では、一旦、 $W00 \Rightarrow W59$ まで生成して、暗号化の時と逆の順番 $W59 \Rightarrow W00$ で拡大鍵を生成することが可能である。これにより、復号化時もすべての拡大鍵をメモ

備える。

【0098】ラウンドファンクション部のRound Key加算処理で用いられるRound Keyは、鍵長256ビットの場合では、Initial Round KeyとRound Key01~Round Key14の合計15個必要となる。各Round keyは処理ブロック長に対応して128ビットで構成されており、鍵スケジュール部で生成される32ビットの拡大鍵で各Round keyを充当するためには、W00~W59の60個の拡大鍵が必要となる。このような拡大鍵W00~W59は、暗号化時にはW00~W59の順に使用され、復号化時にはW59~W00の順に使用される。この実施形態では、表10に示すように、暗号化時にW00~W59の順に拡大鍵を生成するとともに、復号化時においてW59~W00の順に拡大鍵の生成を行うように構成する。

【0099】

【表10】

りに格納しておく必要はなく、各ラウンドに必要な拡大鍵だけを生成しながら、復号処理を行うことが可能となる。

【0101】まず、ラウンドファンクション部でのRound Key加算用拡大鍵の生成について説明する。表10に示すように各ラウンドのRound Key加算で4個の拡大鍵(32bit単位)が使用され、拡大鍵はラウンドファンクションのMix Column変換 + Round Key加算のバックグラウンドで演算されるため4個の拡大鍵は4サイクルで作成す

ばよい。このため図9に示すような回路構成において、1サイクルで1個の拡大鍵を生成する構成となっている。拡大鍵レジスタ120はシフトレジスタで構成されており、ラウンドファンクション部で使用する現在の拡大鍵は、フリップフロップ121の出力WOKEYを用いる。

【0102】拡大鍵生成ロジック部101のセクタ105 (SEL#B)は、表11に示すように、鍵長と暗号化/復号化の2種類の条件により切り換え制御が行われる。また、拡大鍵生成ロジック部101の出力が供給されるセクタ111, 112, 113 (SEL#E~SEL#G)は、鍵長に基づいて、表12のように設定される。ただし、暗号鍵が初期値として入力される際には、各セクタ111~113のセクタ位置はbが選択される。さらに、セクタ107, 109 (SEL#C~D)は、表13に示すように、各拡大鍵の生成論理により切り換え制御が行われる。ROM102には、EXOR回路104に供給される定数Rcon[i]が格納されており、アドレスi番地に対応す

## SEL\_C~SEL\_D制御

論理		SEL_C	SEL_D
拡大	$W[i]=W[i-Nk]\sim W[i-1]$	*	b
大	$W[i]=W[i-Nk]\sim \text{SubByte}(W[i-1])$	b	a
鍵	$W[i]=W[i-Nk]\sim \text{SubByte}(\text{RotByte}(W[i-1]))\sim \text{Rcon}[i/Nk]$	a	a
ByteSub		a	b

\*: don't care

【0106】

【表14】

Rcon ROMテーブル

Rcon_Addr	Hex	Bin
01	0x01	0000_0001
02	0x02	0000_0010
03	0x04	0000_0100
04	0x08	0000_1000
05	0x10	0001_0000
06	0x20	0010_0000
07	0x40	0100_0000
08	0x80	1000_0000
09	0x1B	0001_1011
10	0x36	0011_0110

【0107】表10に示すように、鍵長が256ビットである場合の各演算について回路動作を説明する。なお、ラウンドファンクション動作前に、鍵入力レジスタ131の各レジスタk0~k7の値をロードすることによって、No.00~07からの初期値が拡大鍵レジスタ120の各フリップフロップ121~128にセットされているものとする。

【0108】暗号化時における拡大鍵W08は、表10に示されているように、 $W08=W00\sim \text{SubByte}(\text{RotByte}(W07))\sim \text{Rcon}[1]$ で演算される。この $W08=W00\sim \text{SubByte}(\text{RotByte}(W07))\sim \text{Rcon}[1]$ の演算開始時には、フリップフロップ121の出力WOKEYにはW00がセットされており、EXOR回路104に入力される。また、フリップフロップ12

る各定数Rcon[i]が表14に示すように格納されている。

【0103】

【表11】

## SEL\_B制御

鍵長	暗号化	復号化
128bit	W3KEY	W1KEY
192bit	W5KEY	W1KEY
256bit	W7KEY	W1KEY

10 【0104】

【表12】

## SEL\_E~SEL\_G制御

鍵長	SEL_E	SEL_F	SEL_G
128bit	a	b	b
192bit	b	a	b
256bit	b	b	a

【0105】

【表13】

30

8の出力W7KEYにはW07がセットされており、このW07がセクタ105 (SEL#B)に入力されている。

【0109】ROM102のRconアドレスを"1"にし、AND回路103に入力される読出信号RCON#ENをイネーブルにして、 $\text{Rcon}[1]\sim W00$ の演算をEXOR回路104で行い、その結果をEXOR回路110に入力する。一方、セクタ105 (SEL#B)を通過したW07は、Rot Byte回路106とSub Byte回路107による処理が行われ、Sub Byte (Rot Byte(W07))の演算結果がEXOR回路110に入力される。よって、EXOR回路110では、 $W08=W00\sim \text{SubByte}(\text{RotByte}(W07))\sim \text{Rcon}[1]$ の演算が行われる。

【0110】次に、拡大鍵W09=W01~W08の演算処理を説明する。W09=W01~W08の演算開始時には、フリップフロップ121の出力WOKEYにはW01がセットされており、EXOR回路104にこれが入力されている。フリップフロップ128の出力W7KEYにはW08がセットされており、セクタ105 (SEL#B)に入力されている。AND回路103に入力される読出信号RCON#ENをディセーブルにして、フリップフロップ121から入力されるW01がそのままEXOR回路110に入力されるように設定する。このとき、セクタ109 (SEL#D)ではセクタ位置がbに設定されており、セクタ105 (SEL#B)を通過したW08が、セクタ109を介してそのままEXOR回路110に入力される。

【0111】よって、EXOR回路110では、 $W09=W01\sim W08$ の演算が行われる。W10~W11、W13~W15も同じパスで

50

それぞれ演算が行われる。拡大鍵W12の演算処理を説明する。拡大鍵W12=W04<sup>1</sup>SubByte(W11)で演算され、この演算の開始時には、フリップフロップ121の出力W0KEYにはW04がセットされ、EXOR回路104に輸入されている。また、フリップフロップ128の出力W7KEYにはW11がセットされておりセクタ105 (SEL#B) に輸入されている。AND回路103に輸入される読出信号RCON#ENをディセーブルとして、W04がそのままEXOR回路104に輸入されるように設定する。一方、セクタ107 (SEL#C) ではセクタ位置がbに設定されており、セクタ105 (SEL#B) を通過したW11が、セクタ107 (SEL#C) を介してSub Byte回路108に輸入される。このことにより、Sub Byte回路108によるSub Byte処理が行われ、Sub Byte(W11) の演算結果がEXOR回路110に輸入される。よって、EXOR回路110では、W12=W04<sup>1</sup>Sub Byte(W11)の演算が行われる。

【0112】以上の要領で全ての拡大鍵について演算が行われる。次に、暗号化時の鍵入力レジスタ131の書換えと、暗号化と復号化終了に伴う拡大鍵初期値のセットアップについて説明する。このセットアップ動作は、次の暗号化もしくは復号化に備え、鍵入力レジスタ131に格納されている拡大鍵初期値を拡大鍵レジスタ120に転送する動作である。

【0113】鍵入力レジスタ131にセットされた拡大鍵初期値は、セクタ132 (SEL#A)により32ビット単位でデータ選択が行われ、セクタ113 (SEL#G)のセクタ位置bを経由して拡大鍵レジスタ120にセットされる。拡大鍵レジスタ120は、前述したようにシフトレジスタ構成であり、フリップフロップ128 (FF7) ⇒フリップフロップ127 (FF6) ⇒フリップフロップ126 (FF5) ⇒フリップフロップ125 (FF4) ⇒フリップフロップ124 (FF3) ⇒フリップフロップ123 (FF2) ⇒フリップフロップ122 (FF1) ⇒フリップフロップ121 (FF0) へとシフトして行き、8サイクルで全ての拡大鍵初期値が転送される。なお、セクタ132 (SEL#A) で選択される鍵入力データは鍵入力レジスタ131のレジスタk0, k1, k2, k3, k4, k5, k6, k7の順番である。

【0114】復号化時の鍵入力レジスタ131の書換えに伴う拡大鍵初期値のセットアップについて説明する。表10のように、復号化においては、拡大鍵初期値を暗号化での最終拡大鍵のセット(W59~W52)とする必要がある。鍵入力レジスタ131の書換えにより、鍵入力レジスタ131にセットされたデータを前述した方法で、一旦、拡大鍵レジスタ120に転送し、暗号化の拡大鍵生成論理に従い、最終拡大鍵のセット(W52~W59)まで図9の回路を動作させる。

【0115】この最終拡大鍵のセットの生成期間に、W52の生成時にW52を鍵入力レジスタ131のレジスタk7に転送し、W53の生成時にW53をレジスタk6に転送し、W54

の生成時にW54をレジスタk5に転送し、W55の生成時にW55をレジスタk4に転送し、W56の生成時にW56をレジスタk3に転送し、W57の生成時にW57をレジスタk2に転送し、W58の生成時にW58をレジスタk1に転送し、W59の生成時にW59をk0に転送することにより鍵入力レジスタ131に逆の順番で最終拡大鍵をセットする。さらに、鍵入力レジスタ131の最終拡大鍵のセットを前述した方法で拡大鍵レジスタ120に転送することで復号化時の鍵入力レジスタの書換えに伴う拡大鍵初期値のセットアップが完了するこの後、セクタ105 (SEL#B)、セクタ107 (SEL#C)、セクタ109 (SEL#D)、セクタ111~113 (SEL#E~SEL#G)を表11~表13に示すようなセクタ位置に設定し、復号化に必要な各拡大鍵を順次生成する。〈Byte Sub変換回路の共用化〉上述した鍵スケジュール部のSub Byte処理およびラウンドファンクション部のByte sub変換処理では、ともに32ビット単位でのByte Sub変換処理を実行しているため、これらの処理回路を共用化することが考えられる。

【0116】たとえば、図9に示す鍵スケジュール部に設けられるSub Byte回路108をラウンドファンクション部のByte Sub変換回路と共用化することについて考察。図4に示すラウンドファンクション部における中間レジスタ兼Shift Row変換回路206からByte Sub変換回路207への入力BSINを、図9に示す拡大鍵生成ロジック部101のセクタ107のセクタ位置cに接続する。さらに、拡大鍵生成ロジック部101のSub Byte回路108からの出力を、図4のByte Sub変換回路207の出力BSOUTとしてセクタ203に接続する。

【0117】Sub Byte回路108を用いてByte Sub変換処理を行う場合には、表13に示すように、セクタ107 (SEL#C) のセクタ位置をcとし、セクタ109 (SEL#D) のセクタ位置をbとする。このことにより、拡大鍵生成ロジック部101のSub Byte回路108を用いてラウンドファンクション部のByte Sub変換処理を実行することが可能となる。

〈Byte Sub変換回路〉Byte Sub変換処理は、8ビット単位での逆数演算と行列演算との組み合わせで構成されており、暗号化時には逆数演算を行った後に行列演算を実行し、復号化時には行列演算を行った後に逆数演算を実行する。このようなByteSub変換処理を、暗号化時と復号化時において共通の回路で実施するために、図10に示すような回路を提案する。

【0118】図10に示すByte Sub変換回路391は、復号化用行列演算回路392と、セクタ393、逆数演算回路394、暗号化用行列演算回路395、セクタ396とで構成されている。

【0119】セクタ393は、入力データと復号化用行列演算回路392の出力が入力されておりいずれか一方を逆数演算回路394に輸入するように構成されている。また、セクタ396は、逆数演算回路394の出



力と暗号化用行列演算回路395の出力が入力されており、いずれか一方を出力するように構成されている。

【0120】暗号化時には、セクタ393を入力データ側にし、セクタ396を暗号化用行列演算回路395側にする。また、復号化時には、セクタ393を復号化用行列演算回路392側にし、セクタ396を逆数演算回路394側にする。このことにより、暗号化時

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

【0123】これを展開すると、次に示す数2のように表すことができる。ただし、ここでの“+”は排他的論理和演算で構成される。

$$\begin{aligned} y_0 &= x_0 && + x_4 + x_5 + x_6 + x_7 + 1 \\ y_1 &= x_0 + x_1 && + x_5 + x_6 + x_7 + 1 \\ y_2 &= x_0 + x_1 + x_2 && + x_6 + x_7 \\ y_3 &= x_0 + x_1 + x_2 + x_3 && + x_7 \\ y_4 &= x_0 + x_1 + x_2 + x_3 + x_4 && \\ y_5 &= && x_1 + x_2 + x_3 + x_4 + x_5 + 1 \\ y_6 &= && x_2 + x_3 + x_4 + x_5 + x_6 + 1 \\ y_7 &= && x_3 + x_4 + x_5 + x_6 + x_7 \end{aligned}$$

【0125】また、復号化時における行列演算処理は、次のような数3で表される。

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

【0127】これを同様に展開すると、次に示す数4のように表すことができる。

【0128】

$$\begin{aligned} y_0 &= && x_2 && + x_5 && + x_7 + 1 \\ y_1 &= x_0 && + x_3 && + x_6 && \\ y_2 &= && x_1 && + x_4 && + x_7 + 1 \\ y_3 &= x_0 && + x_2 && + x_5 && \\ y_4 &= && x_1 && + x_3 && + x_6 && \\ y_5 &= && x_2 && + x_4 && + x_7 \\ y_6 &= x_0 && + x_3 && + x_5 && \\ y_7 &= && x_1 && + x_4 && + x_6 \end{aligned}$$

【0129】暗号化用行列演算回路の一例を図11に示す。ここでは、8ビットの入力レジスタ401、出力レジスタ403および排他的論理和と否定で構成される論理回路402で構成される。暗号化時において数2に示すような排他的論理和演算を実行するためには、論理回

におけるByte Sub変換処理と、復号化時におけるByteSub変換処理とを共通の回路構成で実現することが可能となる。

【0121】暗号化時における行列演算処理は、次のような数1で表される。

【0122】

【数1】

【0124】

【数2】

【0126】

【数3】

路402内の排他的論理和回路のうち、重複する演算処理を共用化することで16個のEXORと4個のNOTを組み合わせることにより実現することが可能となる。

【0130】復号化用行列演算回路の一例を図12に示す。暗号化用行列演算回路と同様に、8ビットの入力レジスタ405、出力レジスタ407および排他的論理和と否定で構成される論理回路406で構成されている。ここでも暗号化用行列演算回路と同様に、復号化時において数4に示すような排他的論理和演算を実行するために、論理回路406内の排他的論理和回路のうち、重複する演算処理を共有化することで13個のEXOR回路と2個のNOTを組み合わせることにより実現することが可能となる。

【0131】暗号化用行列演算回路の他の例を図13に示す。この暗号化用行列演算回路は、入力レジスタ411、出力レジスタ414、定数保持用レジスタ413および排他的論理回路で構成される論理回路412を備え

ている。入力レジスタ 411、出力レジスタ 414、定数保持用レジスタ 413は、いずれも 8 ビットのシフトレジスタで構成されており、クロックに同期して 1 ビットずつ右にローテートシフトする。

【0132】数 1 の右辺第 1 項の定数は、1 つの行が 3 つの“0”と 5 つの“1”でなり、1 つずつシフトするように構成されている。これから、入力レジスタ 411 の  $x_0, x_4, x_5, x_6, x_7$  の各ビットをローテートシフトさせながら論理回路 412 に入力して EXOR 演算を実行することで、数 1 の右辺第 1 項の行列演算を行うことができる。

【0133】また、定数保持用レジスタ 413 には、下位ビットから順に数 1 右辺第 2 項の定数がセットされている。定数保持用レジスタ 413 の値をローテートシフトさせながら最下位ビットの値を論理回路 412 に入力し EXOR 演算を実行することで、数 1 右辺第 2 項の行列演算を行うことができる。

【0134】このことにより、入力レジスタ 411 にデータがセットされると、最初のクロックで  $y_0$  が演算されて出力レジスタ 414 に格納される。次のクロックでは、 $y_1$  が演算されて出力レジスタ 414 に格納され、以下順次演算処理を実行して 8 クロックで ( $y_7, y_6, y_5, y_4, y_3, y_2, y_1, y_0$ ) の演算が完了する。この場合の論理回路 412 は 5 個の EXOR 回路で数 2 の演算処理を実行することが可能となる。

【0135】これと同様に構成した復号化用行列演算回路の他の例を図 14 に示す。この復号化用行列演算回路は、入力レジスタ 415、出力レジスタ 418、定数保持用レジスタ 417 および排他的論理回路で構成される論理回路 416 を備えている。入力レジスタ 415、出力レジスタ 418、定数保持用レジスタ 417 は、いずれも 8 ビットのシフトレジスタで構成されており、クロックに同期して 1 ビットずつ右にローテートシフトする。

【0136】数 3 の右辺第 1 項の定数は、1 つの行が 5 つの“0”と 3 つの“1”の組み合わせでなり、1 つずつシフトするように構成されている。これから、入力レジスタ 415 の  $x_2, x_5, x_7$  の各ビットをローテートシフトさせながら論理回路 416 に入力して EXOR 演算を実行することで、数 3 の右辺第 1 項の行列演算を行うことができる。

【0137】また、定数保持用レジスタ 417 には、下位ビットから順に数 3 右辺第 2 項の定数がセットされている。定数保持用レジスタ 417 の値をローテートシフトさせながら最下位ビットの値を論理回路 416 に入力し EXOR 演算を実行することで、数 3 右辺第 2 項の行列演算を行うことができる。

【0138】このことにより、入力レジスタ 415 にデータがセットされると、最初のクロックで  $y_0$  が演算されて出力レジスタ 418 に格納される。以下、暗号化時と同様に順次演算処理を実行して 8 クロックで ( $y_7, y_6, y_5, y_4, y_3, y_2, y_1, y_0$ ) の演算が完了する。この場合の論理

回路 416 は 3 個の EXOR 回路で数 4 の演算処理を実行することが可能となる。

(付記 1) 暗号鍵から所定の処理ブロック長に対応するビット数の複数の Round Key を生成し、入力データと前記 Round Key のうちの 1 つと EXOR 演算を実行する EXOR 演算部と、Byte Sub 変換、Shift Row 変換、Mix Column 変換、Round Key 加算を含む Round 処理を複数回実行する Round 処理部とを備えるラウンドファンクション部により入力データと前記 Round Key による暗号化／復号化処理を前記処理ブロック長毎に実行する暗号回路であって、前記 Round 処理部が、入力データを前記処理ブロック長よりも小さい実行ブロック長に分割する第 1 セクタと、前記実行ブロック長毎に前記 Round Key の値を入力データに加算する第 1 Round Key 加算回路と、前記第 1 Round Key 加算回路の出力を一時的に格納するとともに前記処理ブロック長による Shift Row 変換を実行する中間レジスタ兼 Shift Row 変換回路と、前記中間レジスタ兼 Shift Row 変換回路の値が前記実行ブロック長毎に入力され Byte Sub 変換を実行する Byte Sub 変換回路と、前記中間レジスタ兼 Shift Row 変換回路の値が前記実行ブロック長毎に加算する第 2 Round Key 加算回路と、前記第 2 Round Key 加算回路の出力に対して Mix Column 変換を実行する Mix Column 変換回路と、前記第 1 セクタ、中間レジスタ兼 Shift Row 変換回路、Byte Sub 変換回路、Mix Column 変換回路の出力のうちいずれか 1 つを前記第 2 Round Key 加算回路に出力する第 2 セクタとを備えることを特徴とする暗号回路。

【0139】(付記 2) 前記実行ブロック長は 8 の倍数ビットである付記 1 に記載の暗号回路。

(付記 3) 前記処理ブロック長が 128 ビットであり、前記実行ブロック長が 32 ビットである付記 1 または 2 に記載の暗号回路。

【0140】(付記 4) 前記暗号鍵の鍵長が 128 ビット、192 ビット、256 ビットのうちのいずれかである付記 1 ～ 3 に記載の暗号回路。

【0141】(付記 5) 前記 Byte Sub 変換回路は、入力データに対して行列演算を実行する復号化用行列演算部と、入力データと前記復号化用行列演算部の出力のうちいずれかを出力する第 3 セクタと、前記第 3 セクタから出力されるデータに対して逆数演算を実行する逆数演算部と、前記逆数演算部から出力されるデータに対して行列演算を実行する暗号化用行列演算部と、前記逆数演算部の出力と前記暗号化用行列演算部の出力のうちいずれか一方を出力する第 4 セクタとを備える付記 1 ～ 4 のいずれかに記載の暗号回路。

【0142】(付記 6) 前記復号化用行列演算部と前記暗号化行列演算部は、1 クロックで 8 ビットの演算を行うように EXOR 回路を接続している付記 5 に記載の暗号回路。

【0143】(付記7) 前記復号化行列演算部と前記暗号化行列演算部は、1クロックで1ビットの演算を行うようにEXOR回路を接続している付記5に記載の暗号回路。

【0144】(付記8) 前記中間レジスタ兼Shift Row変換回路に入力されるデータのシフト量に関するシフトデータを暗号化時と復号化時において逆順に入力することにより、前記中間レジスタ兼Shift Row変換回路を暗号化と復号化に共通に使用する、付記1～7のいずれかに記載の暗号回路。

【0145】(付記9) 前記Mix Column変換回路は、乗数固定の複数の乗算器と前記複数の乗算器の排他的論理和を演算するEXOR回路とを備え、各乗算器に入力されるデータと各乗算器に設定された乗数との間で行列演算を実行する、付記1～8のいずれかに記載の暗号回路。

【0146】(付記10) 前記Mix Column変換回路は、8ビット単位で演算が可能な4つの乗算器と前記4つの乗算器の出力に基づいて排他的論理和演算を実行するEXOR回路とを有する演算器を4つ備える付記9に記載の暗号回路。

【0147】(付記11) 前記乗算器が、2つの乗数を制御可能であり暗号化と復号化に共通に使用される付記9または10に記載の暗号回路。

【0148】(付記12) 前記乗算器は、上位ビットからの加算値を制御するように構成される、付記11に記載の暗号回路。

【0149】(付記13) 暗号鍵を前記実行ブロック長に応じたビット数に分割して出力する第5セクタと、前記実行ブロック長毎にデータをラッチするフリップフロップ回路が複数段接続されたシフトレジスタと、前記シフトレジスタの最終段のフリップフロップ回路の出力と定数群のうちから選択される1つの定数との排他的論理和演算を実行する第1EXOR回路と、前記シフトレジスタのフリップフロップのうち暗号化時に演算対象となるフリップフロップの出力と復号化時に演算対象となるフリップフロップの出力とが入力されいづれか1つを選択的に出力する第6セクタと、前記第6セクタの出力をローテーション処理するRot Byte処理回路と、前記第6セクタの出力と前記Rot Byte処理回路の出力とが入力され、いづれか1つを選択的に出力する第7セクタと、前記第7セクタの出力に対して前記実行ブロック長毎のByte Sub変換を実行するSub Byte処理回路と、前記第6セクタの出力と前記Sub Byte処理回路の出力が入力され、いづれか1つを選択的に出力する第8セクタと、前記第1EXOR回路の出力と前記第8セクタの出力とに基づいて排他的論理和演算を実行する第2EXOR回路と、前記シフトレジスタのフリップフロップのうち暗号化時にその出力が演算対象となるフリップフロップに対して、前記第2EXOR回路の出力と、隣接する段のフリ

ップフロップの出力とのうちいづれかを選択的に出力するシフトレジスタ部セクタと、を備える、前記暗号鍵から所定の処理ブロック長に対応するビット数の複数のRound Keyを前記実行ブロック長に対応するビット数毎に分割された拡大鍵として生成する拡大鍵スケジュール回路を有する付記1～12のいずれかに記載の暗号回路。

【0150】(付記14) 前記シフトレジスタは、32ビット単位でデータ処理を実行する8個のフリップフロップ回路を備え、前記第6セクタは、前記フリップフロップのうち下から2段目、4段目、6段目、8段目のフリップフロップの出力が入力され、いづれか1つを出力するように構成されている付記13に記載の暗号回路。

【0151】(付記15) 前記第7セクタに前記中間レジスタ兼Shift Row変換回路の出力が入力され、前記Sub Byte処理回路の出力を前記第2セクタに入力することにより、前記Sub Byte処理回路と前記Round処理部のByte Sub変換回路とを共用する、付記13または14に記載の暗号回路。

【0152】

【発明の効果】本発明によれば、特定処理回路の処理データを所定の実行ブロック長に細分化することにより、AESブロック暗号アルゴリズムをコンパクトな回路で実行可能となる。また、暗号化時における処理回路と復号化時における処理回路を共有化するとともに、鍵スケジュール部とラウンドファンクション部の回路の一部を共有化することでさらに回路規模を小さくすることが可能となる。

【図面の簡単な説明】

【図1】 RijndaelアルゴリズムによるAESの処理ブロック図。

【図2】 鍵スケジュールのプログラムリスト。

【図3】 想定される回路実装の例を示すブロック図。

【図4】 本発明の1実施形態に採用されるラウンドファンクション部のブロック図。

【図5】 中間レジスタ兼Shift Row変換回路のブロック図。

【図6】 Mix Column変換回路のブロック図。

【図7】 乗算器の構成を示すブロック図。

【図8】 乗算器の他の構成を示すブロック図。

【図9】 鍵スケジュール部のブロック図。

【図10】 Byte Sub変換回路のブロック図。

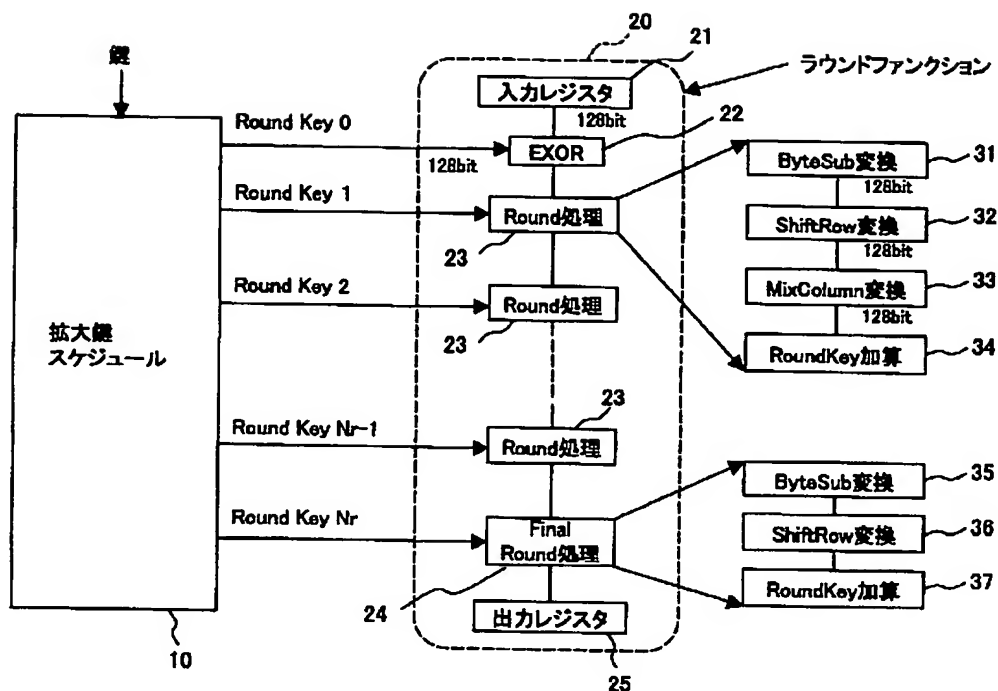
【図11】 暗号化用行列演算回路のブロック図。

【図12】 復号化用行列演算回路のブロック図。

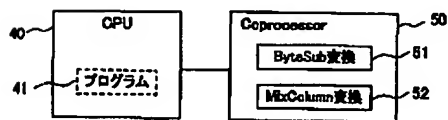
【図13】 暗号化用行列演算回路の他の例を示すブロック図。

【図14】 復号化用行列演算回路の他の例を示すブロック図。

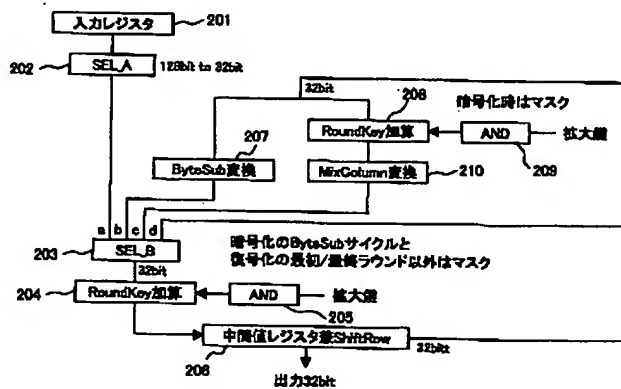
【図 1】



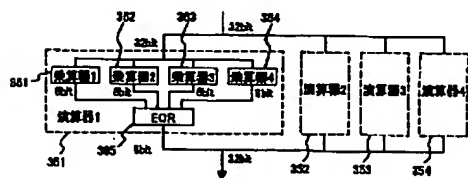
【図 3】



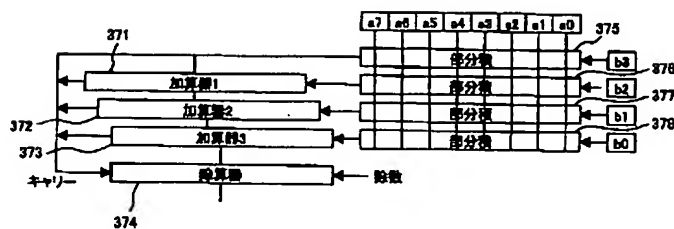
【図 4】



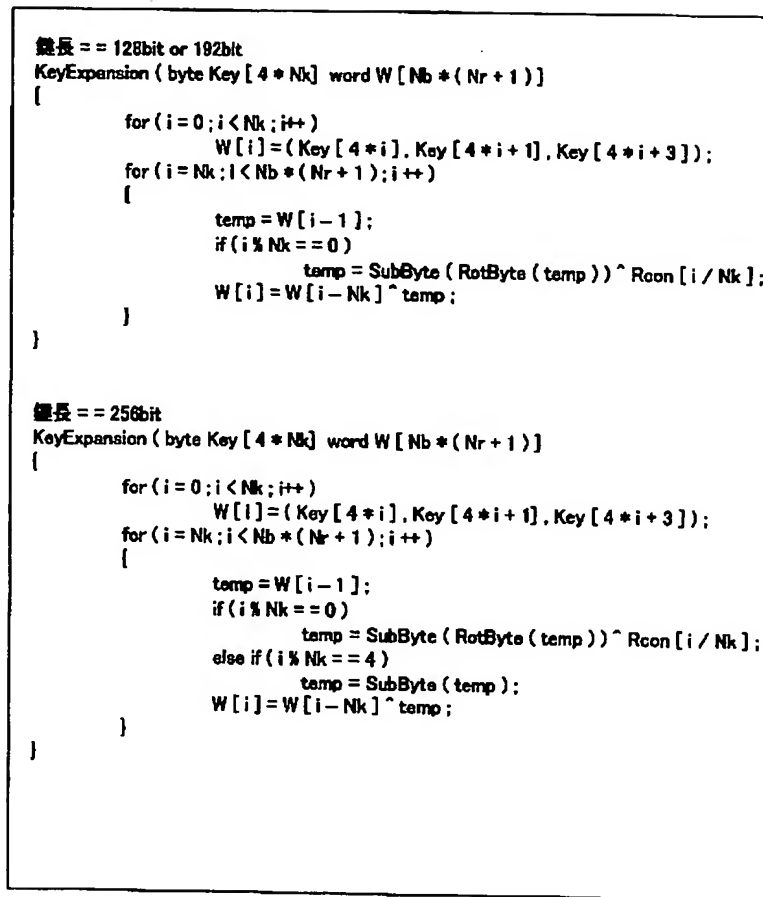
【図 6】



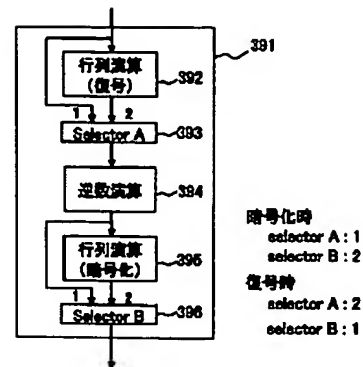
【図 7】



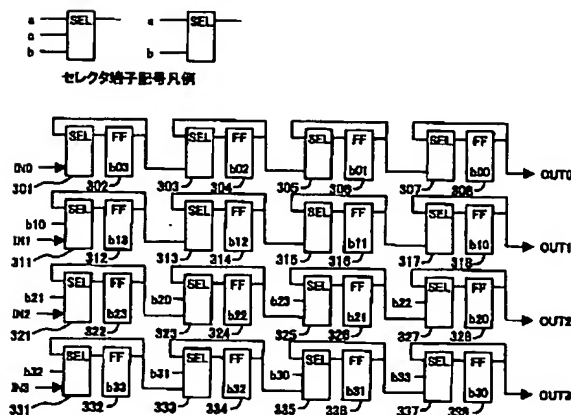
【図2】



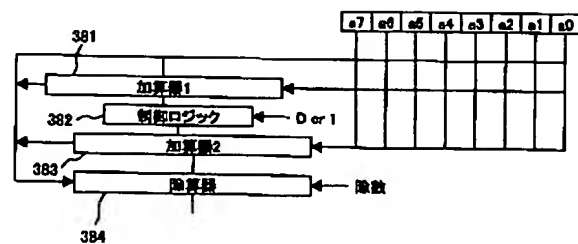
【図10】



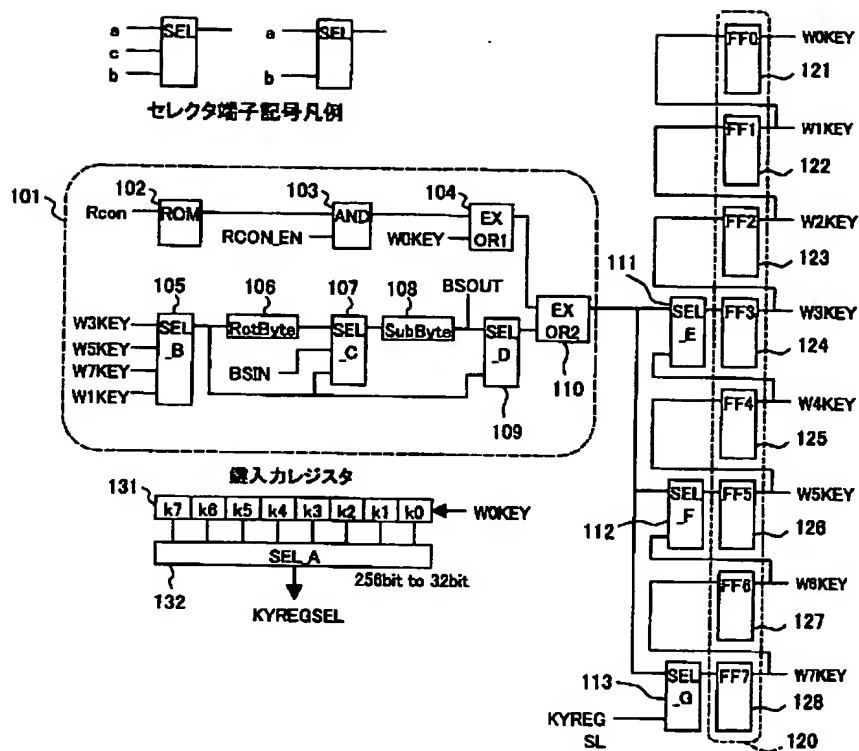
【図5】



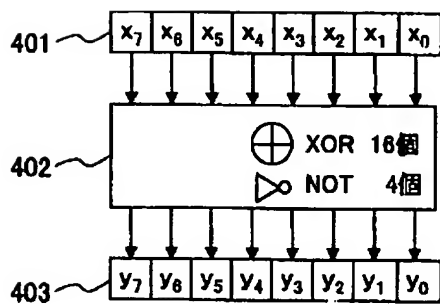
【図8】



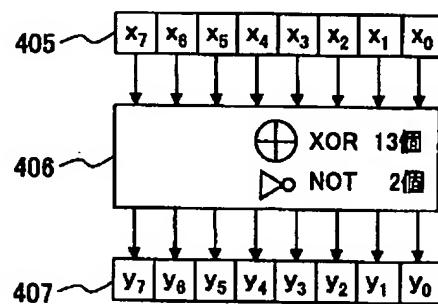
【图 9】



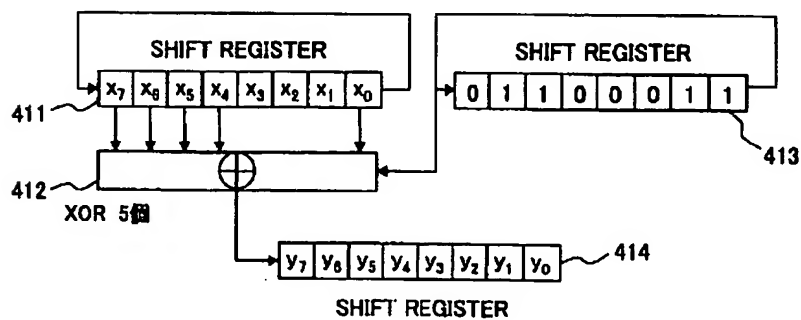
【図 1 1】



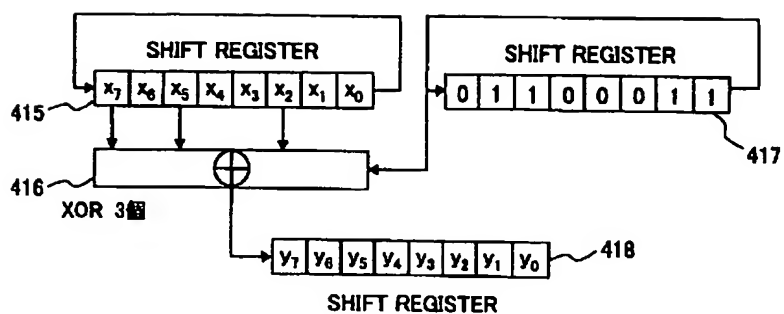
【图 12】



【図 13】



【図 14】



フロントページの続き

(72)発明者 林 朋弘  
神奈川県横浜市港北区新横浜 2 丁目15番16  
号 株式会社富士通コンピュータテクノロ  
ジ内

(72)発明者 出口 千佳広  
神奈川県横浜市港北区新横浜 2 丁目15番16  
号 株式会社富士通コンピュータテクノ  
ジ内

(72)発明者 藤原 由実  
神奈川県横浜市港北区新横浜 2 丁目15番16  
号 株式会社富士通コンピュータテクノ  
ジ内

Fターム(参考) 5J104 AA18 JA03 NA02 NA22



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**